L 5

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 000000000        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 000000000        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 000000000        AAAAAAAAA
SSS              YYY         YYY  SSS              LLL              000         000  AAA         AAA
SSS              YYY         YYY  SSS              LLL              000         000  AAA         AAA
SSS              YYY         YYY  SSS              LLL              000         000  AAA         AAA
SSS                 YYY   YYY     SSS              LLL              000         000  AAA         AAA
SSS                 YYY   YYY     SSS              LLL              000         000  AAA         AAA
SSS                 YYY   YYY     SSS              LLL              000         000  AAA         AAA
   SSSSSSSSS           YYY           SSSSSSSSS     LLL              000         000  AAA         AAA
   SSSSSSSSS           YYY           SSSSSSSSS     LLL              000         000  AAA         AAA
   SSSSSSSSS           YYY           SSSSSSSSS     LLL              000         000  AAA         AAA
            SSS        YYY                    SSS  LLL              000         000  AAAAAAAAAAAAAAA
            SSS        YYY                    SSS  LLL              000         000  AAAAAAAAAAAAAAA
            SSS        YYY                    SSS  LLL              000         000  AAAAAAAAAAAAAAA
            SSS        YYY                    SSS  LLL              000         000  AAA         AAA
            SSS        YYY                    SSS  LLL              000         000  AAA         AAA
            SSS        YYY                    SSS  LLL              000         000  AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     000000000     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     000000000     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     000000000     AAA         AAA
```

```
_$2

Sym
---
SS1
SS1

SS1
SS1
SS1
SS1
SS1
SS1
SYS
SYS
SYS
TRY
UNL
WRI
```

```
**FILE**ID**ERRSUB780                              G 9



EEEEEEEEEE  RRRRRRRR    RRRRRRRR      SSSSSSSS  UU      UU  BBBBBBBB    77777777      888888      000000
EEEEEEEEEE  RRRRRRRR    RRRRRRRR      SSSSSSSS  UU      UU  BBBBBBBB    77777777      888888      000000
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  88      88  00      00
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  88      88  00      00
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  88      88  00    0000
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  88      88  00    0000
EEEEEEEE    RRRRRRRR    RRRRRRRR      SSSSSS    UU      UU  BBBBBBBB          77      888888    00  00  00
EEEEEEEE    RRRRRRRR    RRRRRRRR      SSSSSS    UU      UU  BBBBBBBB          77      888888    00  00  00
EE          RR  RR      RR  RR              SS  UU      UU  BB      BB      77      88      88  0000    00
EE          RR  RR      RR  RR              SS  UU      UU  BB      BB      77      88      88  0000    00
EE          RR    RR    RR    RR            SS  UU      UU  BB      BB    77        88      88  00      00      ....
EE          RR    RR    RR    RR            SS  UU      UU  BB      BB    77        88      88  00      00      ....
EEEEEEEEEE  RR      RR  RR      RR  SSSSSSSS    UUUUUUUUUU  BBBBBBBB    77            888888      000000        ....
EEEEEEEEEE  RR      RR  RR      RR  SSSSSSSS    UUUUUUUUUU  BBBBBBBB    77            888888      000000        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .NOSHOW CONDITIONALS
0000            .TITLE  ERRSUB780 - ERROR SUBROUTINES FOR VAX 11/780
0000
0000
0000
0000
0000
0000            .IDENT  'V04-002'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ;++
0000    ;
0000    ; FACILITY:
0000    ;
0000    ;       EXECUTIVE,  LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000    ;
0000    ; AUTHOR:
0000    ;
0000    ;       N. KRONENBERG, JULY 2, 1979.
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V04-003 WMC00001        Wayne Cardoza           13-Sep-1984
0000    ;               CRD reporting must not be turned off for VENUS.
0000    ;
0000    ;       V04-002 CWH4002         CW Hobbs                08-Sep-1984
0000    ;               Correct typo in TCM0010, use "-" instead of "="
0000    ;
0000    ;       V04-001 TCM0010         Trudy C. Matthews       07-Sep-1984
0000    ;               For the venus processor: move turning on cache from routine
```

```
0000    ;               EXE$INIPROCREG to a new routine: INI$CACHE.  Correct the
0000    ;               order in which registers are saved on the stack in EXE$REGSAVE.
0000    ;
0000    ;       V03-022 TCM0009         Trudy C. Matthews       30-Jul-1984
0000    ;               When turning off CRD interrupts in EXE$INIPROCREG for VENUS,
0000    ;               read the processor register and write it back to preserve
0000    ;               the state of other bits in the register.
0000    ;
0000    ;       V03-021 TCM0008         Trudy C. Matthews       23-Jul-1984
0000    ;               Remove venus code that queries the console for how to set up
0000    ;               cache and FBOX state.  Instead always turn the cache and
0000    ;               FBOX on (and let the normal error handling code turn it off
0000    ;               if its bad).
0000    ;
0000    ;       V03-020 DWT0214         David W. Thiel          02-May-1984
0000    ;               Revise MicroVAX I TODR register simulation.
0000    ;
0000    ;       V03-019 KDM0096         Kathleen D. Morse       27-Mar-1984
0000    ;               Add missing indirection in MicroVAX I memory CSR
0000    ;               CRD enabling.
0000    ;
0000    ;       V03-018 KPL0101         Peter Lieberwirth       4-Mar-1984
0000    ;               Add extra vectors now defined in SYSLOAVEC.  These vectors
0000    ;               are insurance for v4.x
0000    ;
0000    ;       V03-017 KPL0100         Peter Lieberwirth       12-Feb-1984
0000    ;               Change RPB$B_BOOTNDT to RPB$W_BOOTNDT, since BI devices
0000    ;               will have 16-bit device types.
0000    ;
0000    ;       V03-016 KDM0092         Kathleen D. Morse       23-Jan-1984
0000    ;               Correct the number of cpu-specific IPRs logged for the
0000    ;               11/730 and MicroVAX I cpus.
0000    ;
0000    ;       V03-015 CWH8001         CW Hobbs                5-Dec-1983
0000    ;               Add entry points for EXE$READP_TODR and EXE$WRITEP_TODR
0000    ;               to access physical TODR register for Nautilus CPU.  For
0000    ;               other processors, these amount to duplicate labels on
0000    ;               EXE$READ_TODR and EXE$WRITE_TODR.
0000    ;
0000    ;       V03-014 KTA3088         Kerbey T. Altmann       17-Oct-1983
0000    ;               Fix bug in 730 conditional for EXE$INIBOOTADP.
0000    ;
0000    ;       V03-013 KDM0081         Kathleen D. Morse       13-Sep-1983
0000    ;               Create Micro-VAX I version.
0000    ;
0000    ;       V03-012 KDM0055         Kathleen D. Morse       12-Jul-1983
0000    ;               Move IPR PME into the cpu-dependent register save and
0000    ;               restore routines.
0000    ;
0000    ;       V03-011 KDM0049         Kathleen D. Morse       07-Jul-1983
0000    ;               Add the following processor registers to the cpu-specific
0000    ;               dump IPRs routine:  ICR, TODR, ACCS.  Add usage of
0000    ;               register:  EXE$READ_TODR and EXE$WRITE_TODR.
0000    ;
0000    ;       V03-010 KDM0048         Kathleen D. Morse       07-Jul-1983
0000    ;               Add loadable routines for referencing the time-of-day
0000    ;               clock:  EXE$READ_TODR, EXE$WRITE_TODR.
```

```
0000  ;
0000  ;       V03-009 TCM0007         Trudy C. Matthews       02-Jun-1983
0000  ;               Fix routine SYSL$CLRSBIA so that it calculates the address
0000  ;               of SBI adapter register space correctly.
0000  ;
0000  ;       V03-008 TCM0006         Trudy C. Matthews       9-Feb-1983
0000  ;               Store enable/disable state of 11/790 cache and FBOX in
0000  ;               EXE$GB_CPUDATA cell during system initialization.
0000  ;
0000  ;       V03-007 TCM0005         Trudy C. Matthews       11-Jan-1983
0000  ;               Add routine SYSL$CLRSBIA.  Add SBIA register initialization
0000  ;               to EXE$INIPROCREG.  Add 11/790 machine check handler to
0000  ;               EXE$TEST_CSR.  Change 11/780 machine check handler to
0000  ;               write PR$_SBIFS back to itself to clear error bits.
0000  ;               Add labels for two "extra" routines, that can be patched
0000  ;               if extra vectors from SYS to SYSLOA are needed in between
0000  ;               major releases.  Make EXE$DUMPCPUREG log the SBI registers
0000  ;               from the SBI the 11/790 system disk is on.
0000  ;
0000  ;       V03-006 TCM0004         Trudy C. Matthews       3-Jan-1983
0000  ;               Add more 11/790-specific code.
0000  ;
0000  ;       V03-005 TCM0003         Trudy C. Matthews       17-Dec-1982
0000  ;               Add conditional assembly switch to the invocations
0000  ;               of 11/790-specific definition macros.
0000  ;
0000  ;       V03-004 TCM0002         Trudy C. Matthews       15-Dec-1982
0000  ;               Added 11/790-specific code to EXE$INIPROCREG.
0000  ;
0000  ;       V03-003 TCM0001         Trudy C. Matthews       13-Dec-1982
0000  ;               Added 11/790-specific code to power down/power up
0000  ;               routines.
0000  ;
0000  ;       V03-002 KTA3018         Kerbey T. Altmann       30-Oct-1982
0000  ;               Remove CI and UBA routines to another module.
0000  ;
0000  ;--
```

```
                0000
                0000    ;
                0000    ; MACRO LIBRARY CALLS:
                0000    ;
                0000
                0000            $ADPDEF                         ;DEFINE ADAPTER OFFSETS
                0000            $BQODEF                         ;DEFINE BOOT QIO OFFSETS
                0000            $BTDDEF                         ;DEFINE BOOT DEVICE TYPES
                0000            $EMBCRDEF                       ;DEFINE ERROR MSG BUFFER OFFSETS
                0000            $IDBDEF                         ;DEFINE INTERRUPT DISPACH OFFSETS
                0000            $IPLDEF                         ;DEFINE INTERRUPT PRIORITY LEVELS
                0000            $MBADEF                         ;DEFINE MASSBUS ADAPTER OFFSETS
                0000            $NDTDEF                         ;DEFINE NEXUS DEVICE TYPES
                0000            $PRDEF                          ;DEFINE INTERNAL PROCESSOR REGISTERS
                0000            $RPBDEF                         ;DEFINE RESTART PARAM BLOCK OFFSETS
                0000            $SSDEF                          ;DEFINE SYSTEM STATUS CODES
                0000            $UBADEF                         ;DEFINE UNIBUS ADAPTER OFFSETS
                0000
                0000            $PR780DEF                       ;DEFINE 11/780 INTERNAL PROCESSOR REGS
                0000
                0000
                0000
                0000    ;
                0000    ; EQUATED SYMBOLS:
                0000    ;
      00000001  0000            C780_LIKE = 1
      00000000  0000            C750_LIKE = 0
                0000
                0000
                0000
                0000
                0000
                0000    ;
                0000    ; Define labels for two "extra" routines.  This reserves some vectors from
                0000    ; SYS.EXE into SYSLOAxxx.EXE that can be patched if another routine must
                0000    ; be added in between major releases.
                0000    ;
                0000    EXE$EXTRA1::                            ; aligned
                0000    EXE$EXTRA2::                            ;  aligned
                0000    EXE$EXTRA3::                            ;    aligned
                0000    EXE$EXTRA4::                            ;      aligned
                0000    EXE$EXTRA5::                            ;        aligned
                0000    EXE$EXTRA6::                            ; packed
                0000    EXE$EXTRA7::                            ;  packed
                0000    EXE$EXTRA8::                            ;    packed
                0000    EXE$EXTRA9::                            ;      packed
                0000    EXE$EXTRA10::                           ;        packed (think this is enough?)
                0000
            00  0000            HALT                            ; Error if these labels are used.
```

```
                      0001   257         .SBTTL  EXE$INIBOOTADP - INITIALIZE THE BOOT DEVICE ADAPTER
                      0001   258 ;+
                      0001   259 ; EXE$INIBOOTADP - GET THE SYSTEM BOOT DEVICE ADAPTER AND INIT IT.
                      0001   260 ;       THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE BOOTDRIVER IS CALLED.
                      0001   261 ;
                      0001   262 ; INPUTS:
                      0001   263 ;
                      0001   264 ;       R6 = RPB ADDRESS
                      0001   265 ;
                      0001   266 ; OUTPUTS:
                      0001   267 ;
                      0001   268 ;       R0-R2 DESTROYED
                      0001   269 ;       OTHER REGISTERS PRESERVED
                      0001   270 ;-
                      0001   271
                  00000000   272         .PSECT  SYSLOA,LONG
                      0000   273         .ENABLE LSB
                      0000   274
                      0000   275 EXE$INIBOOTADP::                           ;SUBROUTINE ENTRY
                      0000   277
        66 A6     91  0000   278         CMPB    RPB$B_DEVTYP(R6),-         ;IS BOOT DEVICE THE CONSOLE
        40 8F         0003   279                 #BTD$K_CONSOLE             ;BLOCK STORAGE DEVICE?
           67     13  0005   280         BEQL    40$                        ;YES, RETURN
   50   60 A6     D0  0007   281         MOVL    RPB$L_ADPVIR(R6),R0        ;GET ADDR OF ADAPTER REG SPACE
                      000B   282
                      000B   284
 52   00A1 C6  03  AB 000B   285         BICW3   #3,RPB$W_BOOTNDT(R6),R2 ;GET GENERIC ADAPTER TYPE
        38     52  B1 0011   286         CMPW    R2,#NDT$_CI                ;CI ADAPTER?
               21  13 0014   287         BEQL    20$                        ;YES, RETURN
        20     52  B1 0016   288         CMPW    R2,#NDT$_MB                ;MASS BUS ADAPTER?
               1D  12 0019   289         BNEQ    INI_UBADP                  ;BRANCH IF NOT
               02  D0 001B   290         MOVL    #MBA$M_CR_ABORT,-          ;ABORT ACTIVE TRANSFER
           04 A0      001D   291                 MBA$L_CR(R0)               ;
                      001F   292
        51     1B  DB 001F   294         MFPR    #PR780$_TODR,R1            ;GET CURRENT TIME (10 MS UNITS)
                      0022   296
                      0022   300
   51   64 A1     9E  0022   304         MOVAB   100(R1),R1                 ;ALLOW ONE SECOND
        08 A0     D5  0026   305 10$:    TSTL    MBA$L_SR(R0)               ;WAIT UNTIL TRANSFER
           08     18  0029   306         BGEQ    15$                        ; IS COMPLETE
                      002B   307
        52     1B  DB 002B   309         MFPR    #PR780$_TODR,R2            ;GET CURRENT TIME
                      002E   311
                      002E   315
                      002E   319
        52     51  D1 002E   320         CMPL    R1,R2                      ;CHECK FOR INTERVAL EXPIRED
               F3  1A 0031   321         BGTRU   10$                        ;NOT YET, WAIT SOME MORE
               01  D0 0033   322 15$:    MOVL    #MBA$M_CR_INIT,-           ;NOW INIT MBA
           04 A0      0035   323                 MBA$L_CR(R0)               ;
                   05 0037   324 20$:    RSB                                ;DONE
                      0038   325
                      0038   327
                      0038   331
                      0038   332 INI_UBADP:                                 ;INIT UBA
                      0038   333
               01  D0 0038   335         MOVL    #UBA$M_CR_INIT,-
           04 A0      003A   336                 UBA$L_CR(R0)               ;INIT UBA
        00010000 8F   D3  003C  25$:    BITL    #UBA$M_CSR_UBIC,-
                 60       0042                  UBA$L_CSR(R0)           ;WAIT FOR UBA INIT
                 F7   13  0043                  BEQL    25$                     ; TO COMPLETE
                          0045
                          0045
                          0045  ;
                          0045  ; CHECK THE VMB VERSION NUMBER.  IF IT EXISTS AND IF IT IS 7 OR GREATER, THEN
                          0045  ; SEE IF ANY UNIBUS MAP REGISTERS TO DISABLE.
                          0045  ;
                          0045
           52    34 A6    D0  0045          MOVL    RPB$L_IOVEC(R6),R2      ;PICK UP THE IOVECTOR FROM RPB
           51    10 A2    B2  0049          MCOMW   BQO$W_VERSION(R2),R1    ;GET VMB VERSION NUMBER 1'S COMPLEMENTED
           12 A2    51    B1  004D          CMPW    R1,BQO$W_VERSION+2(R2)  ;CHECK AGAINST CHECK WORD IN VMB
                    1B    12  0051          BNEQ    40$                     ;IF NOT, ASSUME NO VERSION NUMBER
           07    10 A2    B1  0053          CMPW    BQO$W_VERSION(R2),#7    ;VERSION 7 OR GREATER OF VMB?
                    15    1F  0057          BLSSU   40$                     ;NO, DON'T BOTH WITH UMR'S
           52    24 A2    D0  0059          MOVL    BQO$L_UMR_DIS(R2),R2    ;GRAB THE NUMBER OF UMR'S TO DISABLE
                    0F    13  005D          BEQL    40$                     ;NONE, LEAVE
                              005F
    04 A0     52    16    78  005F          ASHL    #22,R2,UBA$L_CR(R0)     ;SET THE UMR DISABLE BITS
                              0064
                              0064  ;
                              0064  ; THIS CODE IS EXECUTED FOR ALL PROCESSORS.  ITS DISABLES ANY UNIBUS MAP
                              0064  ; REGISTERS ASSOCIATED WITH UNIBUS MEMORY TO PREVENT CONTENTION BETWEEN
                              0064  ; SBI AND UNIBUS ADDRESSES.
                              0064  ;
                              0064
           51    0800 C0  DE  0064          MOVAL   UBA$L_MAP(R0),R1        ;ADDRESS OF FIRST REGISTER
                       81  D4  0069  30$:    CLRL    (R1)+                   ;DISABLE IT
                    FB 52  F5  006B          SOBGTR  R2,30$                  ;LOOP UNTIL ALL DONE
                           05  006E  40$:    RSB                             ;DONE WITH UBA INIT
                               006F          .DISABLE LSB
                          006F              .SBTTL  EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                          006F              .SBTTL  EXE$STARTUPADP - STARTUP ANY ADAPTERS
                          006F      ;+
                          006F      ; EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                          006F      ;       THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE DUMP IS TAKEN TO
                          006F      ;       ENSURE THAT ALL ADAPTERS THAT NEED TO BE QUIESENT ARE.
                          006F      ;
                          006F      ; INPUTS:
                          006F      ;
                          006F              IPL = 31
                          006F      ;
                          006F      ; OUTPUTS:
                          006F      ;
                          006F              OTHER REGISTERS PRESERVED
                          006F      ;-
                          006F              .ENABLE LSB
                          006F
                          006F      EXE$STARTUPADP::
              17     BB   006F              PUSHR   #^M<R0,R1,R2,R4>          ; Save a register
     51    B6'AF     DE   0071              MOVAL   B^ADP_TBL_UP,R1           ; Address of startup table
              06     11   0075              BRB     5$                        ; Join common code
                          0077
                          0077      EXE$SHUTDWNADP::
              17     BB   0077              PUSHR   #^M<R0,R1,R2,R4>          ; Save a register
     51    9E'AF     DE   0079              MOVAL   B^ADP_TBL_DWN,R1          ; Address of shutdown table
     FFFFFFFC'9F     DE   007D      5$:     MOVAL   @#<IOC$GL_ADPLIST- -
              52          0083                          ADP$L_[INK>,R2        ; Get pointer to head of adapter list
     52    04 A2     D0   0084      10$:    MOVL    ADP$L_LINK(R2),R2         ; Flink onward
              11     13   0088              BEQL    20$                       ; Branch if at end of list
        54    62     D0   008A              MOVL    ADP$L_CSR(R2),R4          ; Get address of CSR
     50    0E A2     3C   008D              MOVZWL  ADP$W_ADPTYPE(R2),R0      ; Get adapter type code
      50    6140     DE   0091              MOVAL   (R1)[R0],R0               ; Get table entry of adap shutdown
         00 B040     16   0095              JSB     @(R0)[R0]                 ; Call adapter shutdown
              E9     11   0099              BRB     10$                       ; Next adapter
                          009B
              17     BA   009B      20$:    POPR    #^M<R0,R1,R2,R4>
              05          009D      30$:    RSB
                          009E
                          009E      ;
                          009E      ; Table of addresses of adapter shutdown routines ordered
                          009E      ; by adapter type in ADP$W_ADPTYPE.
                          009E      ;
                          009E
                          009E      ADP_TBL_DWN:                              ; Address table start
          FFFFFFFF        009E              .LONG   30$-.                     ; 0-MBA
          FFFFFFFB        00A2              .LONG   30$-.                     ; 1-UBA
          FFFFFFF7        00A6              .LONG   30$-.                     ; 2-DR32
          FFFFFFF3        00AA              .LONG   30$-.                     ; 3-MA780
          FFFFFF52'       00AE              .LONG   CI$SHUTDOWN-.             ; 4-CI
          FFFFFFEB        00B2              .LONG   30$-.                     ; Rsvrd for future expansion
                          00B6
                          00B6      ;
                          00B6      ; Table of addresses of adapter startup routines ordered
                          00B6      ; by adapter type in ADP$W_ADPTYPE.
                          00B6      ;
                          00B6
                          00B6      ADP_TBL_UP:                               ; Address table start
```

```
FFFFFF4A' 00B6   452        .LONG   MBA$INITIAL-.           ; 0-MBA
FFFFFF46' 00BA   453        .LONG   UBA$INITIAL-.           ; 1-UBA
FFFFFFDF  00BE   454        .LONG   30$-.                   ; 2-DR32
FFFFFF3E' 00C2   455        .LONG   MA$INITIAL-.            ; 3-MA780
FFFFFFD7  00C6   456        .LONG   30$-.                   ; 4-CI
FFFFFFD3  00CA   457        .LONG   30$-.                   ; Rsvrd for future expansion
          00CE   458
          00CE   459        .DISABLE LSB
```

```
                    00CE              .SBTTL  EXE$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
                    00CE      ;+
                    00CE      ; DUMP CPU-SPECIFIC IPR'S INTO ERROR MESSAGE BUFFER.
                    00CE      ;
                    00CE      ; TWENTY-FOUR LONGWORDS ARE RESERVED IN THE EMB FOR CPU-SPECIFIC
                    00CE      ; IPR'S.  THE FORMATS FOR VARIOUS CPU'S ARE:
                    00CE      ;
                    00CE      ; 11/780:        11/750:         11/730:         11/790:                  uVAX I:
                    00CE      ;
                    00CE      ; ICR            ICR             ICR             ICR                      UNUSED(0)
                    00CE      ; TODR           TODR            TODR            TODR                     APPROX TODR
                    00CE      ; ACCS           ACCS            ACCS            ACCS                     UNUSED(0)
                    00CE      ; SBIFS          TBDR            21 UNUSED(0)    SBISTS  (1st SBI)        21 UNUSED(0)
                    00CE      ; SBISC          CADR                            SILOCMP      ''
                    00CE      ; SBIMT          MCESR                           MAINT        ''
                    00CE      ; SBIER          CAER                            SBIERR       ''
                    00CE      ; SBIS           CMIERR                          TMOADDRS     ''
                    00CE      ; 16 SBI SILO    16 UNUSED(0)                    16 SBI SILO  ''
                    00CE      ;
                    00CE      ; INPUTS:
                    00CE      ;
                    00CE      ;       R0 - ADDR IN EMB OF START OF CPU-SPECIFIC REGISTERS=
                    00CE      ;            OFFSET EMB$L_CR_CPUREG
                    00CE      ;
                    00CE      ; OUTPUTS:
                    00CE      ;
                    00CE      ;       R0,R1 DESTROYED
                    00CE      ;       ALL OTHER REGISTERS PRESERVED
                    00CE      ;-
                    00CE
                    00CE              .ENABL  LSB
                    00CE
                    00CE      EXE$DUMPCPUREG::                                ;SUBROUTINE ENTRY
                    00CE
                    00CE
80    1A     DB     00CE              MFPR    #PR780$_ICR,(R0)+               ;LOG INTERVAL COUNT REG.
80    1B     DB     00D1              MFPR    #PR780$_TODR,(R0)+              ; TIME-OF-DAY REG.
80    28     DB     00D4              MFPR    #PR780$_ACCS,(R0)+              ; ACCELERATOR CONTROL REG.
80    30     DB     00D7              MFPR    #PR780$_SBIFS,(R0)+             ; SBI FAULT REG.
80    32     DB     00DA              MFPR    #PR780$_SBISC,(R0)+             ; SBI COMPARATOR REG
80    33     DB     00DD              MFPR    #PR780$_SBIMT,(R0)+             ; SBI MAINT REG.
80    34     DB     00E0              MFPR    #PR780$_SBIER,(R0)+             ; SBI ERROR REG.
80    31     DB     00E3              MFPR    #PR780$_SBIS,(R0)+              ; SBI TIMEOUT REG.
51    10     D0     00E6              MOVL    #16,R1                          ;GET # SILO ENTRIES TO DUMP
80    31     DB     00E9      10$:    MFPR    #PR780$_SBIS,(R0)+              ;DUMP SILO TO EMB
   FA 51     F5     00EC              SOBGTR  R1,10$                          ;
                    00EF
                    00EF
                    00EF
                    00EF
                    00EF
                    00EF
                    00EF
                    00EF
                    00EF      90$:
             05     00EF              RSB
                    00F0              .DISABLE LSB
```

```
                                 00F0   575
```

```
                    00F0           .SBTTL  EXE$READ_TODR (P) - READ TIME-OF-DAY CLOCK
                    00F0    ;+
                    00F0    ; READS THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                    00F0    ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                    00F0    ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                    00F0    ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                    00F0    ;
                    00F0    ; INPUTS:
                    00F0    ;
                    00F0    ;       NONE.
                    00F0    ;
                    00F0    ; OUTPUTS:
                    00F0    ;
                    00F0    ;       R0 - TODR VALUE
                    00F0    ;       ALL OTHER REGISTERS PRESERVED
                    00F0    ;-
                    00F0
                    00F0    EXE$READP_TODR::                            ; SUBROUTINE ENTRY
                    00F0
                    00F0           ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                    00F0           ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                    00F0           ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                    00F0           ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                    00F0           ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$READ_TODR ENTRY
                    00F0           ; WHICH WILL FABRICATE THE TIME FROM THE QUADWORD SYSTEM TIME.
                    00F0
                    00F0                                                ; NOT NAUTILUS - FALL THROUGH TO READ_TODR
                    00F0
                    00F0    EXE$READ_TODR::                             ; SUBROUTINE ENTRY
                    00F0
                    00F0
   50    1B    DB   00F0           MFPR    #PR780$_TODR,R0              ; TODR IS A PROCESSOR REGISTER.
                    00F3
                    00F3
                    00F3
                    00F3
                    00F3
                    00F3
                    00F3
                    00F3
               05   00F3           RSB
                    00F4
                    00F4              .SBTTL  EXE$WRITE_TODR (P) - WRITES TIME-OF-DAY CLOCK
                    00F4      ;+
                    00F4      ; WRITES THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                    00F4      ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                    00F4      ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                    00F4      ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                    00F4      ;
                    00F4      ; INPUTS:
                    00F4      ;
                    00F4      ;       R0 - CONTAINS VALUE TO BE WRITTEN INTO TODR
                    00F4      ;
                    00F4      ; OUTPUTS:
                    00F4      ;
                    00F4      ;       NEW TIME VALUE WRITTEN INTO TODR.
                    00F4      ;       ALL REGISTERS PRESERVED.
                    00F4      ;-
                    00F4
                    00F4      EXE$WRITEP_TODR::                         ; SUBROUTINE ENTRY
                    00F4
                    00F4              ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                    00F4              ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                    00F4              ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                    00F4              ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                    00F4              ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$WRITE_TODR ENTRY
                    00F4              ; WHICH WILL FABRICATE A NEW QUADWORD SYSTEM TIME.
                    00F4
                    00F4                                                ; NOT NAUTILUS - FALL THROUGH TO WRITE_TODR
                    00F4
                    00F4      EXE$WRITE_TODR::                          ; SUBROUTINE ENTRY
                    00F4
                    00F4
 1B    50    DA     00F4              MTPR    R0,#PR780$_TODR           ; TODR IS A PROCESSOR REGISTER.
                    00F7
                    00F7
                    00F7
                    00F7
                    00F7
                    00F7
                    00F7
                    00F7
                    00F7
             05     00F7              RSB
```

```
                 00F8                .SBTTL  EXE$REGSAVE - SAVE CPU-SPECIFIC IPR'S
                 00F8        ;+
                 00F8        ; EXE$REGSAVE - CALLED BY POWERFAIL TO SAVE CPU-SPECIFIC IPR'S ON
                 00F8        ;               THE STACK
                 00F8        ;
                 00F8        ; INPUTS: NONE
                 00F8        ;
                 00F8        ; OUTPUTS:
                 00F8        ;
                 00F8        ;               R0 DESTROYED
                 00F8        ;               OTHER GENERAL REGISTERS PRESERVED
                 00F8        ;               IPR'S SAVED ON THE STACK AS FOLLOWS:
                 00F8        ;
                 00F8        ;          11/780:         11/750:          11/730:           11/790:   uVAX I:
                 00F8        ;
                 00F8        ; 0(SP)   PME              PME              PME               ACCS      (none)
                 00F8        ; 4(SP)   SBIMT            TBDR                               CSWP
                 00F8        ; 8(SP)                    CADR                               PME
                 00F8        ;
                 00F8        ;-
                 00F8
                 00F8                .ENABL  LSB
                 00F8
                 00F8        EXE$REGSAVE::                                  ;SUBROUTINE ENTRY
         01   BA 00F8                POPR    #^M<R0>                        ;CLEAR RETURN FROM STACK
                 00FA
                 00FA
  7E     3D   DB 00FA                MFPR    #PR780$_PME,-(SP)              ;SAVE PERFORMANCE MONITOR ENABLE
  7E     33   DB 00FD                MFPR    #PR780$_SBIMT,-(SP)            ;SAVE SBI MAINT REG
                 0100
                 0100
                 0100
                 0100
                 0100
                 0100
                 0100
         60   17 0100                JMP     (R0)                           ;DONE, RETURN
                 0102
                 0102
                 0102                .DSABL  LSB
```

```
                    0102            .SBTTL  EXE$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
                    0102    ;+
                    0102    ; EXE$REGRESTOR - CALLED BY POWERFAIL RECOVERY TO RESTORE CPU-SPECIFIC
                    0102    ;                IPR'S FROM THE STACK.
                    0102    ;
                    0102    ; INPUTS:
                    0102    ;
                    0102    ;                       R6 - TOP OF STACK
                    0102    ;                       STACK SET UP AS DEFINED IN OUTPUTS OF EXE$REGSAVE.
                    0102    ;
                    0102    ; OUTPUTS:
                    0102    ;
                    0102    ;                       R0 DESTROYED
                    0102    ;                       OTHER GENERAL REGISTERS PRESERVED
                    0102    ;                       CPU-SPECIFIC IPR'S RESTORED FROM STACK
                    0102    ;                       R6 - ADDRESS OF 1ST CPU-INDEPENDENT SAVED IPR
                    0102    ;
                    0102    ;-
                    0102
                    0102            .ENABL  LSB
                    0102
                    0102    EXE$REGRESTOR::                                 ;SUBROUTINE ENTRY
          01    BA  0102            POPR    #^M<R0>                         ;CLEAR RETURN FROM STACK
                    0104
                    0104
     33   86    DA  0104            MTPR    (R6)+,#PR780$_SBIMT             ;RESTORE SBI MAINT REGISTER
     3D   86    DA  0107            MTPR    (R6)+,#PR780$_PME               ;RESTORE PERFORMANCE MONITOR ENABLE
                    010A
                    010A
                    010A
                    010A
                    010A
                    010A
                    010A
          60    17  010A            JMP     (R0)                            ;DONE, RETURN
                    010C
                    010C            .DSABL  LSB
```

```
                                  010C   846            .SBTTL  EXE$INIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
                                  010C   847  ;+
                                  010C   848  ; EXE$INIPROCREG - PERFORM INITIALIZATION OF INTERVAL TIMER AND
                                  010C   849  ;         CPU-DEPENDENT REGISTERS. CALLED FROM INIT AND POWERFAIL.
                                  010C   850  ;
                                  010C   851  ; INPUTS:
                                  010C   852  ;
                                  010C   853  ;         NONE
                                  010C   854  ;
                                  010C   855  ; OUTPUTS:
                                  010C   856  ;
                                  010C   857  ;         NONE
                                  010C   858  ;-
                                  010C   859
                                  010C   860 EXE$INIPROCREG::                           ; INIT PROCESSOR REGISTERS
                                  010C   861
                                  010C   864                                            ; FOR 11/780, CONTINUE
                                  010C   865                                            ;
                 00'     E1       010C   866            BBC     S^#EXE$V_CRDENABL,-
    07 00000000'9F                010E   867                    @#EXE$GL_FLAGS,10$         ; IF CLR, IGNORE CRD ERRORS
34     0000C000 8F       DA       0114   868            MTPR    #<3@14>,S^#PR780$_SBIER ; SET CRD INTERRUPT ENABLE
                 00'     E1       011B   869 10$:       BBC     S^#EXE$V_SBIERR,-
    07 00000000'9F                011D   870                    @#EXE$GL_FLAGS,20$         ; IF CLR, IGNORE SBI ERRORS
30     00040000 8F       DA       0123   871            MTPR    #<1@18>,S^#PR780$_SBIFS ; SET SBI FAULT ENABLE
                                  012A   872
                 00'     E0       012A   925 20$:       BBS     S^#EXE$V_NOCLOCK,-
    0E 00000000'9F                012C   926                    @#EXE$GL_FLAGS,30$         ; BRANCH IF NOT USING CLOCK
                                  0132   927
                                  0132   931
19     FFFFD8F0 8F       DA       0132   933            MTPR    #-<10*1000>,S^#PR780$_NICR ; LOAD NEXT INTERVAL REGISTER
                                  0139   935
                                  0139   939
                                  0139   943
18     800000D1 8F       DA       0139   944            MTPR    #^X800000D1,S^#PR$_ICCS ; CLEAR ERROR AND START CLOCK
                         05       0140   945 30$:       RSB                             ; AND RETURN
                                  0141   946
                                  0141   962
```

```
        0141            .SBTTL  SYSL$CLRSBIA
        0141   ;++
        0141   ; SYSL$CLRSBIA  - ON 11/790, CLEAR SBIA ERROR REGISTERS
        0141   ;                - ON 11/780, 11/750, 11/730, AND MICRO-VAX I, THIS IS A NOP
        0141   ;
        0141   ; THIS ROUTINE IS CALLED TO CLEAR OUT SBIA ERROR BITS AFTER A MACHINE CHECK
        0141   ; OCCURS (WHEN MACHINE CHECK IS HANDLED LOCALLY).
        0141   ;
        0141   ; THIS ROUTINE SHOULD BE CALLED AT IPL 31.
        0141   ;
        0141   ; INPUTS:
        0141   ;       ABUS_TYPE       - AN ARRAY TYPE CODES; IDENTIFIES EACH ADAPTER ON THE
        0141   ;                         ABUS.
        0141   ;       ABUS_VA         - AN ARRAY OF ADAPTER SPACE VA'S FOR EACH ADAPTER
        0141   ;                         ON THE ABUS.
        0141   ;
        0141   ; OUTPUTS:
        0141   ;       SBI ERROR BITS ARE CLEARED FOR EACH SBIA ON THE ABUS.
        0141   ;       ALL REGISTERS PRESERVED.
        0141   ;++
        0141   SYSL$CLRSBIA::
    05  0141           RSB                             ; AND RETURN
```

```
                         0142         .SBTTL  EXE$TEST_CSR
                         0142  ;+
                         0142  ; EXE$TEST_CSR - TEST A UNIBUS CONTROLLER CSR FOR EXISTENCE
                         0142  ;
                         0142  ; THIS TEST IS CPU-DEPENDENT.  THE FOLLOWING CPU'S ARE SUPPORTED:
                         0142  ;
                         0142  ;      11/780  -TEST CSR AND CHECK RESULT IN THE UBA STATUS REGISTER.
                         0142  ;      11/750  -NON-EXISTENT CSR IS REPORTED VIA MACHINE CHECK AS A
                         0142  ;               NON-EXISTENT MEMORY REFERENCE.  CONNECT A TEMPORARY
                         0142  ;               MACHINE CHECK HANDLER, TEST THE CSR, AND RESTORE THE
                         0142  ;               ORIGINAL MACHINE CHECK HANDLER.
                         0142  ;      11/730  -ACTION IS THE SAME AS FOR THE 11/750.
                         0142  ;      11/790  -ACTION IS THE SAME AS FOR THE 11/780.
                         0142  ;      MICRO-VAX I -ACTION IS SAME AS FOR THE 11/750.
                         0142  ;
                         0142  ; THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH OR JUMP TO SUBROUTINE AT IPL 31.
                         0142  ;
                         0142  ; INPUTS:
                         0142  ;
                         0142  ;      R0 = CSR ADDRESS
                         0142  ;      R6 = ADAPTER CONFIGURATION REGISTER ADDRESS
                         0142  ;
                         0142  ; OUTPUTS:
                         0142  ;
                         0142  ;      R0 LOW BIT SET/CLEAR FOR EXISTENT/NONEX CSR
                         0142  ;      OTHER REG:STERS PRESERVED.
                         0142  ;-
                         0142
                         0142         .ENABL  LSB
                         0142
                         0142  EXE$TEST_CSR::                          ;SUBROUTINE ENTRY
                         0142
              06    BB   0142         PUSHR   #^M<R1,R2>              ;SAVE REGISTERS
                         0144
                         0144  ;
                         0144  ; This next line of code is present so that this routine continues to function
                         0144  ; correctly when the UNIBUS adapter is powered down.  Moving 0 into the UBA
                         0144  ; Status Register has no effect when addressing the actual adapter register,
                         0144  ; and clears out any garbage bits in memory when UNIBUS space is re-mapped to
                         0144  ; the "black hole" page.
                         0144  ;
       08 A6    00    D0 0144         MOVL    #0,UBA$L_SR(R6)         ;WHEN UBA IS REMAPPED
51  00000000'GF       D0 0148         MOVL    G^EXE$GL_SCB,R1         ;GET SCB ADDRESS
           04 A1      DD 014F         PUSHL   4(R1)                   ;SAVE CURRENT MCHECK HANDLER ADDR
        52    5E      D0 0152         MOVL    SP,R2                   ;MARK CURRENT STACK POSITION
  04 A1      68'AF    DE 0155         MOVAL   B^MCHK_780,4(R1)        ;CONNECT TEMP 11/780 MCHECK HANDLER
              60      B5 015A         TSTW    (R0)                    ;ATTEMPT TO READ CSR
  08 A6      08 A6    D0 015C         MOVL    UBA$L_SR(R6),UBA$L_SR(R6) ;CLEAR AND CHECK FOR ERROR
              1B      12 0161         BNEQ    NONEX_DEV               ;BRANCH IF ERROR
        50    01      9A 0163  OK:    MOVZBL  #SS$_NORMAL,R0          ;SET STATUS TO SUCCESS
              18      11 0166         BRB     TEST_DONE               ;JOIN COMMON EXIT
                         0168
                         0168  ;
                         0168  ; TEMPORARY CSR TEST MACHINE CHECK HANDLER FOR THE 11/780:
                         0168  ;
                         0168
                         0168         .ALIGN  LONG
```

```
                 0168   1094 MCHK_780:
   7E    30   DB 0168   1095         MFPR    S^#PR780$_SBIFS,-(SP)   ;GET SBI FAULT STATUS REGISTER
00 6E    19   E5 016B   1096         BBCC    #25,(SP),10$            ;CLEAR ERROR 1ST PASS BIT
   30    8E   DA 016F   1097 10$:    MTPR    (SP)+,S^#PR780$_SBIFS   ;WRITE BACK TO CLEAR THE FAULT
50    04 AE   D0 0172   1098         MOVL    4(SP),R0                ;PICK UP SUMMARY PARAMETER
   5E    52   D0 0176   1099         MOVL    R2,SP                   ;CLEAR MCHECK FRAME OFF STACK
   05    50   D1 0179   1100         CMPL    R0,#5                   ;IS IT READ DATA SUBSTITUTE?
         E5   13 017C   1101         BEQL    OK                      ;YES, THEN IT IS READ W/BAD PARITY
                 017E   1103
                 017E   1118
                 017E   1188 NONEX_DEV:                              ;
         50   D4 017E   1189         CLRL    R0                      ;SET STATUS TO FAILURE
                 0180   1190 TEST_DONE:
      04 A1 8ED0 0180   1191         POPL    4(R1)                   ;RESTORE SYSTEM MCHECK HANDLER
                 0184   1192 TEST_DONE_2:
         06   BA 0184   1193         POPR    #^M<R1,R2>              ;RESTORE REGISTERS
              05 0186   1194         RSB                             ;RETURN RESULT TO CALLER
                 0187   1195         .DISABLE LSB
```

```
                             0187          .SBTTL  ADPLINK - LINK ADAPTER CONTROL BLOCK INTO ADP LIST
                             0187    ;+
                             0187    ; ADPLINK LINKS THE ADAPTER CONTROL BLOCK TO THE END OF THE ADP LIST
                             0187    ;
                             0187    ; INPUT:
                             0187    ;       R2 - ADDRESS OF NEW ADP
                             0187    ; OUTPUTS:
                             0187    ;       ADP IS LINK TO THE END OF THE ADPLIST LOCATED BY IOC$GL_ADPLIST.
                             0187    ;       R0,R1 destroyed.
                             0187    ;-
                             0187
                             0187    ADPLINK::                                       ;
50   FFFFFFFC'9F       9E    0187            MOVAB   @#<IOC$GL_ADPLIST-ADP$L_LINK>,R0
                             018E                                                    ; START OF LIST
        51    04 A0    D0    018E    10$:    MOVL    ADP$L_LINK(R0),R1               ; FLINK TO FIRST ENTRY
                 05    13    0192            BEQL    20$                             ; AT END
           50    51    D0    0194            MOVL    R1,R0                           ; TRY AGAIN
                 F5    11    0197            BRB     10$                             ;
        04 A0    52    D0    0199    20$:    MOVL    R2,ADP$L_LINK(R0)               ; CHAIN NEW ADP TO END OF LIST
                       05    019D            RSB                                     ; AND RETURN
                             019E
                             019E            .END
```

```
ADP$L_CSR                       = 00000000
ADP$L_LINK                      = 00000004
ADP$W_ADPTYPE                   = 0000000E
ADPLINK                           00000187 RG      03
ADP_TBL_DWN                       0000009E R       03
ADP_TBL_UP                        000000B6 R       03
BQO$L_UMR_DIS                   = 00000024
BQO$W_VERSION                   = 00000010
BTD$K_CONSOLE                   = 00000040
C750_LIKE                       = 00000000
C780_LIKE                       = 00000001
CI$SHUTDOWN                       ********    X    03
CPU_TYPE                        = 00000001
EXE$DUMPCPUREG                    000000CE RG      03
EXE$EXTRA1                        00000000 RG      01
EXE$EXTRA10                       00000000 RG      01
EXE$EXTRA2                        00000000 RG      01
EXE$EXTRA3                        00000000 RG      01
EXE$EXTRA4                        00000000 RG      01
EXE$EXTRA5                        00000000 RG      01
EXE$EXTRA6                        00000000 RG      01
EXE$EXTRA7                        00000000 RG      01
EXE$EXTRA8                        00000000 RG      01
EXE$EXTRA9                        00000000 RG      01
EXE$GL_FLAGS                      ********    X    03
EXE$GL_SCB                        ********    X    03
EXE$INIBOOTADP                    00000000 RG      03
EXE$INIPROCREG                    0000010C RG      03
EXE$READP_TODR                    000000F0 RG      03
EXE$READ_TODR                     000000F0 RG      03
EXE$REGRESTOR                     00000102 RG      03
EXE$REGSAVE                       000000F8 RG      03
EXE$SHUTDWNADP                    00000077 RG      03
EXE$STARTUPADP                    0000006F RG      03
EXE$TEST_CSR                      00000142 RG      03
EXE$V_CRDENABL                    ********    X    03
EXE$V_NOCLOCK                     ********    X    03
EXE$V_SBIERR                      ********    X    03
EXE$WRITEP_TODR                   000000F4 RG      03
EXE$WRITE_TODR                    000000F4 RG      03
INI_UBADP                         00000038 R       03
IOC$GL_ADPLIST                    ********    X    03
MA$INITIAL                        ********    X    03
MBA$INITIAL                       ********    X    03
MBA$L_CR                        = 00000004
MBA$L_SR                        = 00000008
MBA$M_CR_ABORT                  = 00000002
MBA$M_CR_INIT                   = 00000001
MCHK_780                          00000168 R       03
NDT$_CI                         = 00000038
NDT$_MB                         = 00000020
NONEX_DEV                         0000017E R       03
OK                                00000163 R       03
PR$_ICCS                        = 00000018
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
PR780$_ACCS                     = 00000028
PR780$_ICR                      = 0000001A
PR780$_NICR                     = 00000019
PR780$_PME                      = 0000003D
PR780$_SBIER                    = 00000034
PR780$_SBIFS                    = 00000030
PR780$_SBIMT                    = 00000033
PR780$_SBIS                     = 00000031
PR780$_SBISC                    = 00000032
PR780$_TODR                     = 0000001B
RPB$B_DEVTYP                    = 00000066
RPB$L_ADPVIR                    = 00000060
RPB$L_IOVEC                     = 00000034
RPB$W_BOOTNDT                   = 000000A1
SS$_NORMAL                      = 00000001
SYSL$CLRSBIA                      00000141 RG      03
TEST_DONE                         00000180 R       03
TEST_DONE_2                       00000184 R       03
UBA$INITIAL                       ********    X    03
UBA$L_CR                        = 00000004
UBA$L_CSR                       = 00000000
UBA$L_MAP                       = 00000800
UBA$L_SR                        = 00000008
UBA$M_CR_INIT                   = 00000001
UBA$M_CSR_UBIC                  = 00010000
```

```
                                                +----------------+
                                                ! Psect synopsis !
                                                +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC    USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
. BLANK .                       00000001  (     1.)  01 (  1.)  NOPIC    USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$ABS$                           00000000  (     0.)  02 (  2.)  NOPIC    USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
SYSLOA                          0000019E  (   414.)  03 (  3.)  NOPIC    USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG

                                             +------------------------+
                                             ! Performance indicators !
                                             +------------------------+

Phase                     Page faults    CPU Time        Elapsed Time
-----                     -----------    --------        ------------
Initialization                     36    00:00:00.04     00:00:00.83
Command processing                131    00:00:00.43     00:00:04.04
Pass 1                            343    00:00:07.55     00:00:24.25
Symbol table sort                   0    00:00:01.06     00:00:02.56
Pass 2                            133    00:00:01.99     00:00:06.21
Symbol table output                11    00:00:00.06     00:00:00.47
Psect synopsis output               3    00:00:00.02     00:00:00.02
Cross-reference output              0    00:00:00.00     00:00:00.00
Assembler run totals              659    00:00:11.16     00:00:38.38

The working set limit was 1650 pages.
70789 bytes (139 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1058 non-local and 18 local symbols.
1222 source lines were read in Pass 1, producing 16 object records in Pass 2.
20 pages of virtual memory were used to define 19 macros.

                                            +--------------------------+
                                            ! Macro library statistics !
                                            +--------------------------+

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                        10
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      6
TOTALS (all libraries)                                16

1123 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ERRSUB780/OBJ=OBJ$:ERRSUB780 MSRC$:CPUSW780/UPDATE=(ENH$:CPUSW780)+MSRC$:ERRSUB/UPDATE=(ENH$:ERRSUB)+EXECML$/LIB
```

0395 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

ERRSUBUV1
LIS

ERRSUB790
LIS

ERRSUB780
LIS

ERRSUB750
LIS

FALMAN
LIS

ERRSUB730
LIS